ยืน ภู่วรวรรณ

ฟังก์ชันแสดงผลกราฟิกในฟลิปเปอร์

สร้างสรรค์งานกราฟิกด้วยลูกเล่นที่แพรวพราว เพิ่มเสน่ห์ให้กับรายงาน

ฟลิปเปอร์เป็นโปรแกรมสนับสนุนการทำงานของดีเบส ภายในตัวฟลิปเปอร์มีฟังก์ชันต่าง ๆ ให้เรียกใช้ได้มากมาย ฟังก์ชันส่วนใหญ่เน้นการใช้งานการแสดงผลกราฟิกซึ่งเป็นฟังก์ชันที่ดีเบสเองยังขาดอยู่อีกมาก ผู้เขียนโปรแกรมสามารถเรียกฟลิปเปอร์ให้ทำงานเพิ่มขึ้นได้ ในฉบับนี้จึงเสนอรายละเอียดฟังก์ชันต่าง ๆ ของฟลิปเปอร์

กลุ่มฟังก์ชันกำหนดค่าเริ่มต้น

ฟังก์ชันเหล่านี้เป็นฟังก์ชันที่ใช้ในการตั้งระบบฟลิปเปอร์ขณะเริ่มต้นเพื่อใช้กับโปรแกรมคลิปเปอร์ หากเกิดข้อผิดพลาดโปรแกรมจะส่งเสียงเตือนให้ทราบ ฟังก์ชันในกลุ่มนี้ประกอบด้วย FLIP_INIT( ) และ FREE_BUF( )

ค่าพารามิเตอร์ที่กำหนดให้กับระบบโดยใส่ตัวเลขลงในฟังก์ชัน FLIP_INIT (ค่าบัฟเฟอร์) เมื่อโปรแกรมเขียนด้วยภาษาซี กำหนดขนาดของข้อมูลหนึ่งตัวเลขใช้ 8 ไบต์ ดังนั้นถ้าเรากำหนดค่าบัฟเฟอร์ขนาด 8000 ก็จะได้ตัวเลขทั้งสิ้น 750 ตัวเลข หากเป็นการกำหนดตัวแปร X กับ Y จะได้ 375 คู่

ในการให้ค่าตัวเลขจึงขึ้นกับตัวแปรที่มีกี่คอลัมน์และกี่แถว เราคำนวณได้ด้วยหลักการที่กล่าวมาแล้วคือ ตัวเลขที่เก็บในบัฟเฟอร์แต่ละตัวมีค่า 8 ไบต์

สำหรับฟังก์ชัน FREE_BUF( ) เป็นฟังก์ชันที่ใช้สำหรับการเคลียหน่วยความจำที่จองไว้ในบัฟเฟอร์จากคำสั่ง FLIP_INIT

ฟังก์ชัน GRAPHMODE( )

เป็นฟังก์ชันสำหรับบอกฟลิปเปอให้รู้จักกับจอภาพที่กำหนดในตัวแปอาร์กิวเมนต์ ค่าของตัวเลขโหมด 1, 2, 3 4 และ 9 โดยตีความได้ดังตารางที่

ฟังก์ชัน INITDATA( )

ค่าพารามิเตอร์ที่ใส่คือค่าจำนวคอลัมน์ เช่น ต้องการกำหนดข้อมูเฉพาะสองคอลัมน์คือ X กับ Y ใช้ INITDATA (2) แต่ถ้าต้องการคอลัมน์ก็ใช้ INITDATA (4) จำนวคอลัมน์สูงสุดถึง 7 จำนวนแถวขอข้อมูลที่ป้อนเท่ากับจำนวนบัฟเฟอร์กำหนดให้ในการกำหนดบัฟเฟอร์เริ่มต้

ตารางที่ 1
ความหมายของตัวแปร
อาร์กิวเมนต์บอกชนิด
ของจอภาพ

โหมด	ความละเอียดจอภาพ	ฮาร์ดแวร์
1	320×200	CGA, EGA, VGA
2	640×200	EGA, EGA, VGA
3	640×350	EGA, VGA
4	640×480	VGA
9	2 สี	เฮอร์คิวลิส, CGA, EGA, VGA

ฟังก์ชัน HP_SETUP (comport, baud, parity, databit, stop)

ฟังก์ชันตั้งค่าเริ่มต้นของพอร์ตสื่อสาร โดยปกติใช้ต่อกับเครื่องพล็อตกราฟ การตั้งพอร์ตสื่อสารต้องให้ตรงกับค่าที่ตั้งไว้ที่พล็อตเตอร์ มิฉะนั้นจะไม่สามารถส่งข้อมูลกันได้

กลุ่มคำสั่งในการสร้างกราฟ

คำสั่งในกลุ่มนี้เกี่ยวข้องโดยตรงกับการเขียนกราฟ การกำหนดรูปแบบกราฟที่ต้องการ ตัวอย่างฟังก์ชันในกลุ่มนี้เช่น

ฟังก์ชัน SET_AREA (x1, y1, x2, y2)

ฟังก์ชันนี้ใช้อาร์กิวเมนต์ x1, y1, x2, y2 กำหนดพื้นที่การแสดงผล โดยถือว่าบนจอภาพมีขอบเขตของการแสดงผลจาก 0,0 ถึง 100,100 การแบ่งพื้นที่วิธีนี้เพื่อให้การแสดงผลสามารถกำหนดภาพกราฟได้หลายรูปซ้อนกันบนจอภาพเดียว ทำให้ได้กราฟที่สวยงามเหมาะแก่การแสดงผล เช่น ถ้าต้องการให้แสดงผลกราฟเฉพาะครึ่งบนของจอภาพก็ใช้คำสั่ง

```
SET_AREA (0,0,100,50)
```

ฟังก์ชัน SET_XSKIP( ) และ SET_YSKIP( )

ค่าพารามิเตอร์ที่เป็นตัวเลขจะบอกการกำกับแกนว่ามีการกระโดดเว้นระยะอย่างไร เช่นตัวอย่าง SET_YSKIP (2) หมายถึงให้เขียนตัวเลขทางด้านแกน Y เว้นทีละสอง ดังตัวอย่างในรูปที่ 1

ฟังก์ชัน SET_TYPE ( ) และ HISTOGRAM ( )

ฟังก์ชัน SET_TYPE ( ) เป็นฟังก์ชันกำหนดชนิดกราฟ โดยมีพารามิเตอร์สองตัวคือ ตัวแรกเป็นคอลัมน์ของข้อมูลที่จะนำมาพล็อต ตัวที่สองบอกชนิดของกราฟดังต่อไปนี้

0 ไม่ลากเส้น
1 กราฟเส้นตรง
2 จุด
3 กราฟแท่ง
4 กราฟแท่งซ้อน
5 กราฟวงกลม
6 กราฟ HI-LO

ลองดูตัวอย่างกราฟเส้นตรงในรูปที่ 2 ที่กำหนดด้วยคำสั่ง SET

ฟังก์ชัน SET_YSIDE ( ) และ SET_LN_PAT ( )

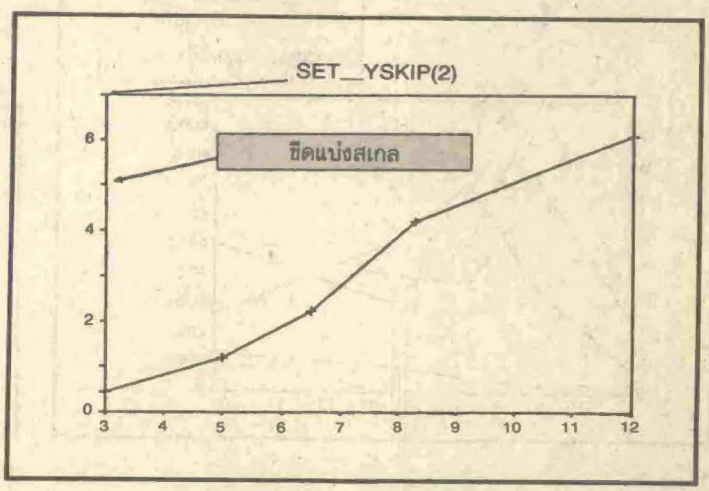
เป็นฟังก์ชันเพื่อกำหนดแกน Y ณ ด้านใดด้านหนึ่งของรูป ค่า 0 หมายถึงทางซ้าย 1 หมายถึงทางขวา เช่น ต้องการให้แกนอ้างอิงสำหรับกราฟเส้นที่ 5 เขียนไว้ทางขวาก็ใช้ SET_YSIDE (5,1) สำหรับการใช้ฟังก์ชัน SET_LN_PAT ( ) คือการกำหนดค่ารูปแบบของเส้นที่จะพล็อต โดยค่าที่กำหนดได้มีดังนี้

0 เส้นทึบ
1 x...x...x...x
2 xxxx...xxxx...
3 xxxxxx...xxxxxx...
4 x.x.x.x.x.
5 xxxx.xxxx.xxxx.
6 xx..xx..xx..xx

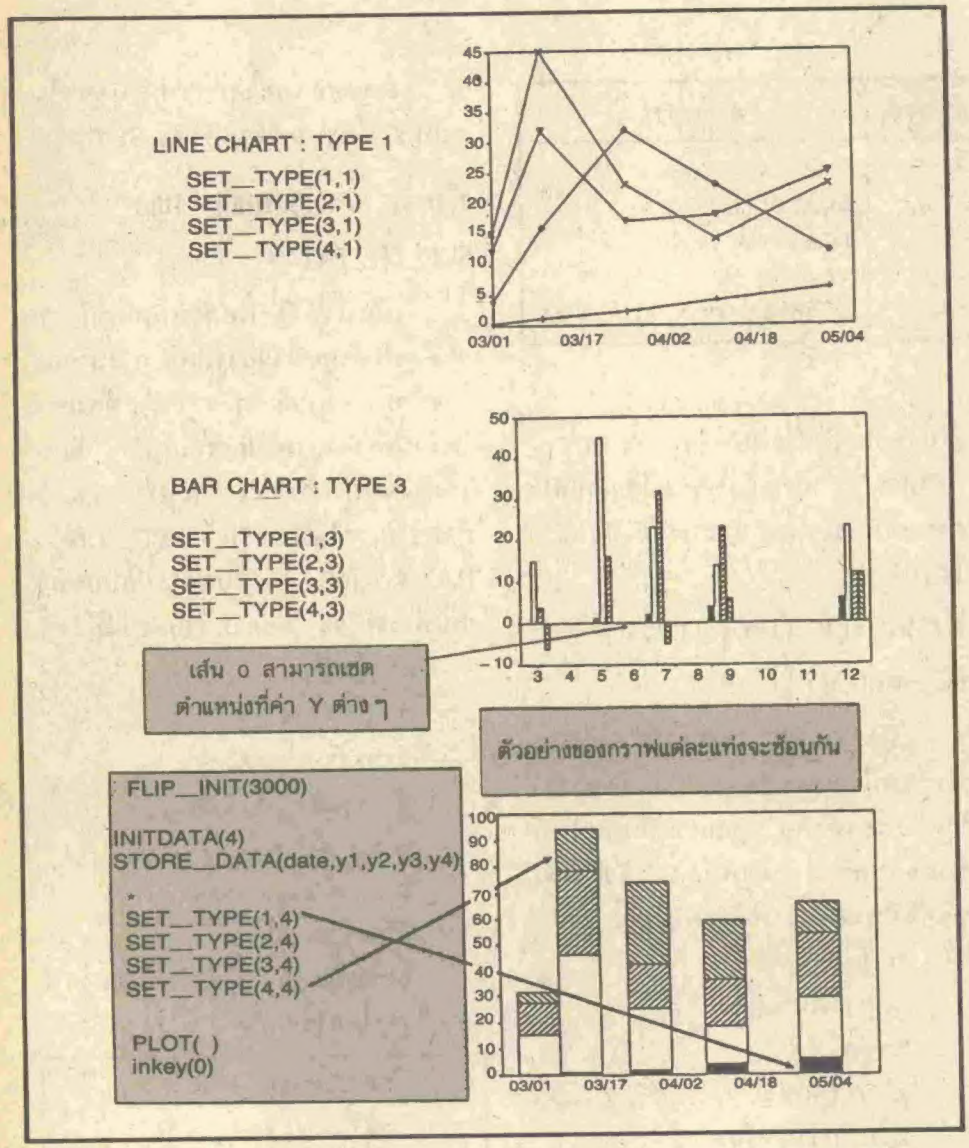
ลองดูตัวอย่างการกำหนดกราฟเส้นที่ 5 โดยใช้คำสั่ง

```
SET_YSIDE (5, 1)
SET_LN_PAT (5, 1)
```

ผลลัพธ์ของกราฟที่ได้บนจอภาพจะเป็นดังรูปที่ 3



รูปที่ 1 ตัวอย่างการใช้ฟังก์ชันกำกับแกน



รูปที่ 2 กราฟเส้นตรงที่กำหนดด้วยคำสั่ง SET

ฟังก์ชัน SET__XLOG( ) และ ฟังก์ชัน SET__YLOG( )

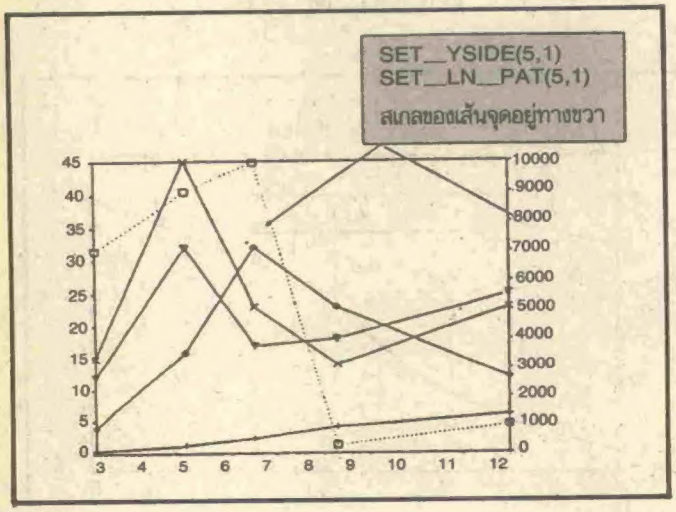
พารามิเตอร์ที่ใช้คือ ON กับ OFF ถ้า ON หมายถึงการเขียนกราฟแกนนั้นเป็นสากลแบบลอการิทึม ค่า ON คือ 1 ค่า OFF คือ 0 ตัวอย่างเช่น

```
ON = 1
OFF = 0
SET__TYPE (1,1)
SET__GRID (1)
SET__LN__PAT (0,1)
SET__YLOG (ON)
```

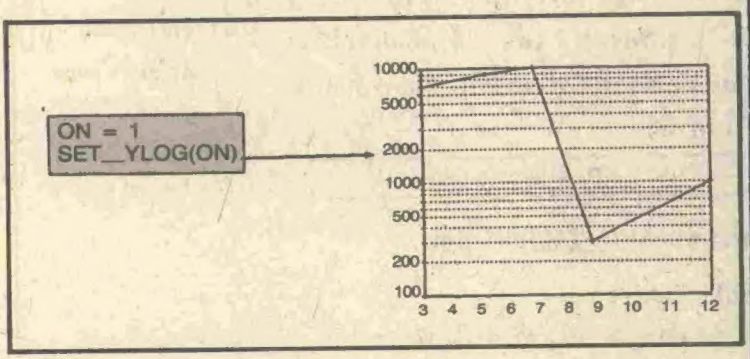
รูปกราฟที่เขียนเสร็จแล้วจะเป็นดังรูปที่ 4

ฟังก์ชัน SET__STYLE( )

เป็นฟังก์ชันที่ใช้สำหรับกำหนดลักษณะของรูปกราฟที่แสดงผลบนจอภาพ ลักษณะกราฟที่กำหนดนี้เป็นพารามิเตอร์ที่กำหนดให้ด้วยตัวเลข เช่น SET__STYLE (1,3) หมายถึงกำหนดให้กราฟตัวแปรแรกมีรูปแบบเป็นขีดจากขวามาซ้าย รูปแบบลักษณะของกราฟที่กำหนดได้คือ กราฟแท่ง กราฟจุด หรือกราฟเส้น ค่าพารามิเตอร์ที่กำหนดตัวที่สองแต่ละค่ามีความหมายดังตารางที่ 2 ลองพิจารณาตัวอย่างในรูปที่ 5



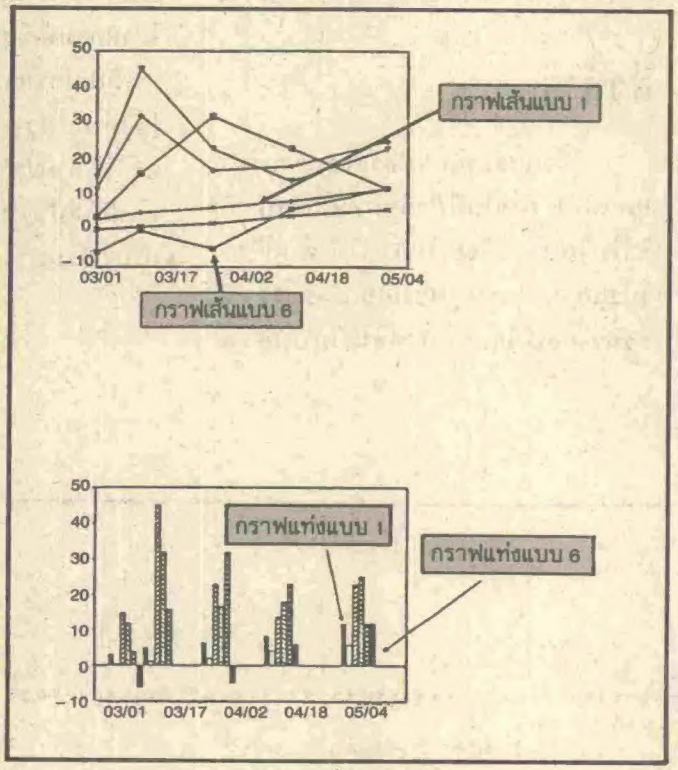
รูปที่ 3



รูปที่ 4 การสร้างกราฟสากลแบบลอการิทึม

งที่ 2
รามิเตอร์ที่กำหนด
ะค่า

ค่าตัวเลข	รูปแบบกราฟแท่ง	วงกลมรูปแบบจุด
0	ไม่มีแท่ง	ไม่มีจุด
1	แท่งทึบ	+
2	กรอบพื้นในขาว	×
3	ลายคาดเฉียงขวาไปซ้าย	สามเหลี่ยม
4	ลายคาดเฉียงซ้ายไปขวา	รูปสี่เหลี่ยมขนมเปียกปูน
5	ลายคาดเฉียงตัดกัน	สี่เหลี่ยม
6	ลายคาดเฉียงขวาไปซ้าย	สี่เหลี่ยมทึบ
7	ลายคาดเฉียงซ้ายไปขวา	จุด
8	ลายคาดเฉียงตัดกัน	เส้นเท่านั้น
9	ไม่มี	วงกลม



5 ผลลัพธ์ที่ได้จากการ
กำหนดลักษณะ
ของกราฟ ด้วยฟังก์ชัน
SET_STYLE

นอกจากนี้ยังมีการกำหนดลักษณะตัวหนังสือและขนาด โดยมีฟังก์ชันกำหนดฟอนต์ ขนาดของฟอนต์ด้วยตัวเลขผ่านเข้าไปยังอาร์กิวเมนต์ของฟังก์ชัน ฟังก์ชันกำหนดฟอนต์ประกอบด้วย

ฟังก์ชัน SET__G__FONT (font__handle)

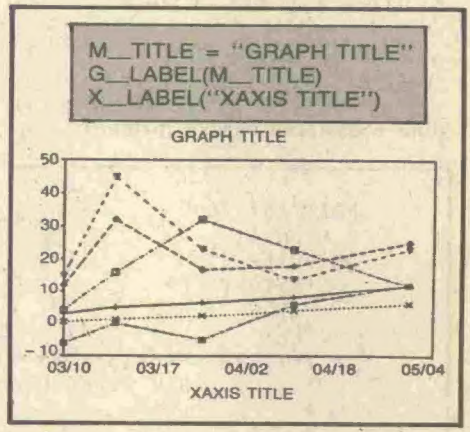
ฟังก์ชัน SET__L__FONT (font__handle)

ฟังก์ชัน SET__S__FONT (font__handle)

ดังตัวอย่างประกอบในรูปที่ 6

กำหนดกริดบนกราฟ

เป็นการลากเส้นกริดบนกราฟทั้งแนวตั้งและแนวนอน โดยกำหนด



รูปที่ 6 ฟังก์ชันกำหนดลักษณะตัวหนังสือและขนาด

ยนลาเบลบนกราฟ

ในการเขียนกราฟมาตรฐานทั่วไป
ารกำหนดข้อความหรือตัวอักษรลง
ำแหน่งต่าง ๆ ของกราฟ ฟลิปเปอร์
หนดฟังก์ชันสำหรับการเขียนตัวอักษร
ำนี้ไว้แล้ว อันประกอบด้วยฟังก์ชัน
ะหน้าที่ต่าง ๆ ดังตารางที่ 3

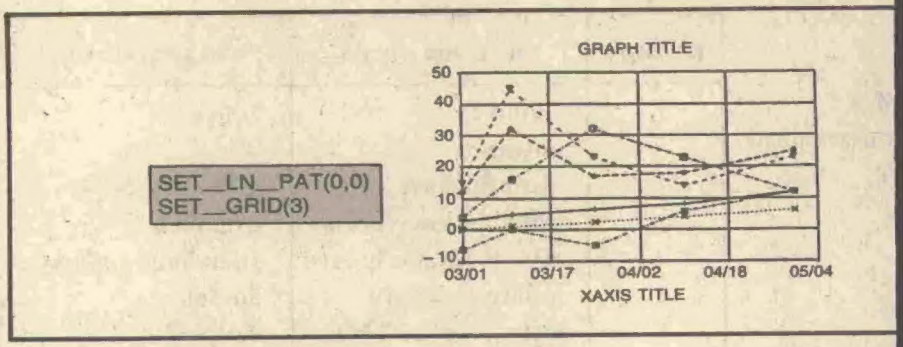
ตารางที่ 3 ฟังก์ชันการเขียนตัวอักษร

ฟังก์ชัน	หน้าที่
ฟังก์ชัน G__LABEL ("Graph Title")	เป็นฟังก์ชันเขียนชื่อกราฟ
ฟังก์ชัน X__LABEL ("Xaxis Title")	เป็นฟังก์ชันเขียนคำอธิบายแกน X
ฟังก์ชัน Y__LABEL ("Yaxis Title")	เป็นฟังก์ชันเขียนคำอธิบายแกน Y
ฟังก์ชัน Y__LABEL2 ("Right Yaxis Title")	เป็นฟังก์ชันเขียนคำอธิบายแกน Y ทางด้านขวา

พารามิเตอร์ในฟังก์ชัน SET_GRID (type) ค่า type ที่กำหนดให้มี 4 ค่า ดังนี้

0 หมายถึงไม่มีกริด
1 หมายถึงกริดในแนวราบ
2 หมายถึงกริดในแนวดิ่ง
3 หมายถึงกริดทั้งสองแนว

ตัวอย่างของการเขียนกริดแสดงไว้ในรูปที่ 7



รูปที่ 7 ตัวอย่างการเขียนกริด

แถมตัวอย่างเล็กน้อย

ฟังก์ชันของฟลิปเปอร์ยังมีอีกมาก ส่วนตัวอย่างในบทความนี้ก็เพื่อให้เห็นจุดเด่นของการใช้งาน การเขียนโปรแกรมในดีเบสและคอมพายล์ด้วยคลิปเปอร์เพื่อลิงก์กับฟังก์ชันภายในฟลิปเปอร์เป็นสิ่งที่ทำให้ผู้เขียนโปรแกรมได้ฟังก์ชันกราฟิกอีกมากมาย การออกรายงานเป็นกราฟิกจึงไม่ใช่เรื่องยากอีกต่อไปแล้ว ในที่นี้จึงขอยกตัวอย่างโปรแกรมที่เขียนด้วยคำสั่งของฟลิปเปอร์ดังรูปที่ 8 และ 9

ส่งท้าย

โปรแกรมฟลิปเปอร์เป็นของบริษัท Prowork ภายในมีฟังก์ชันที่เก็บรวบรวมไว้ในไลบรารีที่จะเรียกมาใช้ได้ ผู้เขียนโปรแกรมควรหาคู่มือฟลิปเปอร์ซึ่งรวบรวมรายละเอียดทุกฟังก์ชัน ในบทความนี้จะกล่าวถึงเพียงบางฟังก์ชันเพื่อแนะนำ ภายในแผ่นโปรแกรมยังมีตัวโปรแกรมที่เขียนให้อีกมากมาย นอกนี้ฟลิปเปอร์ยังมีฟอนต์ตัวอักษรให้ใช้ได้หลายรูปแบบ รวมทั้งโปรแแก้ไขลักษณะรูปแบบตัวอักษรด้วย หาฟลิปเปอร์มาใช้แล้วจะรู้ว่าฟลิปเป็นโปรแกรมที่มีประโยชน์มาก

รูปที่ 8 ตัวอย่างโปรแกรมเขียนกราฟแบบต่าง ๆ

```
FLIP_INIT(600)
M_type=1
DO WHILE .T.
   INITDATA(2)
   CLEAR
   TEXT
   ************************************************************************
   *        SET_TYPE(column,type)                                         *
   *                              column- y column value                  *
   *                              type  - type graph  1-line              *
   *                                                  2-point             *
   *                                                  3-bar               *
   *                                                  4-stacked           *
   *                                                  5-pie               *
   *                                                                      *
   *        The X value is automatically set. RET_XTYPE()  0 - numeric    *
   *                                                       1 - date       *
   *                                                       2 - character  *
   ************************************************************************
   ENDTEXT
*
   @ 19,10 SAY 'X axis type: '+STR(RET_XTYPE(),1,0)
   @ 20,10 SAY 'Y axis type: '+STR(RET_YTYPE(1),1,0)
   @ 22,0 say 'Type: 'GET M_type PICT '9' valid M_type >0 .and. M_type <6
```

ที่ 8 (ต่อ)

```
   READ
*
   IF LASTKEY()=27
      EXIT
   ENDIF
*
   SET_TYPE(1,M_type)
   STORE_DATA('Jan',3)
   STORE_DATA('Feb',5)
   STORE_DATA('Mar',7)
   STORE_DATA('Apr',3)
   STORE_DATA('May',3.51)
   STORE_DATA('Jun',5)
   STORE_DATA('Jul',6.3)
   STORE_DATA('Aug',3)
   STORE_DATA('Sep',4.5)
   STORE_DATA('Oct',8.5)
   STORE_DATA('Nov',10.2)
   STORE_DATA('Dec',11.5)
*
   IE M_type=5
      PIE(1)
   ELSE
      PLOT()
   ENDIF
   inkey(0)
   TEXTMODE()
*
ENDDO
```

ที่ 9 ตัวอย่างโปรแกรมเขียนกราฟแกนออกริทึม

```
FLIP_INIT(600)
HP_SETUP(1,9600,0,8,1)
M_log=0
M_y1=1
m_y2=10000
M_type=1
M_out=0
max_fh=FONT_OPEN("TTL09_33.@")
DO WHILE .T.
   CLEAR
   TEXT
   ************************************************************
   * Example: SET_XLOG()                                      *
   *          SET_YLOG()                                      *
   *                                                          *
   *          Data in buffer:     X          Y                *
   *                             ----       -----            *
   *                               1          3              *
   *                               2         50              *
   *                               5       1000              *
   *                              12       3000              *
   *                              50      10000              *
   *                             123      13000              *
   ************************************************************
   ENDTEXT
```

รูปที่ 9 (ต่อ)

```
   INITDATA(2)
   RESET()
*
   @ 18,0 say 'Output: 'GET M_out PICT '9'
   @ 19,0 say 'Type: 'GET M_type PICT '9'
   @ 20,0 say 'Log: (0-numeric scale, 1-X log, 2-Y log, 3-both log): 'GET M_log PICT '9'
   READ
*
   IF LASTKEY()=27
      EXIT
   ENDIF
*
   SET_XLOG(0)
   SET_YLOG(0)
   DO CASE
      CASE M_log=1
         SET_XLOG(1)
      CASE M_log=2
         SET_YLOG(1)
      CASE M_log=3
         SET_XLOG(1)
         SET_YLOG(1)
   ENDCASE
*
   SET TYPE(1,M_type)
*
   STORE_DATA(1,3)
   STORE_DATA(2,50)
   STORE_DATA(5,1000)
   STORE_DATA(12,3000)
   STORE_DATA(50,10000)
   STORE_DATA(123,13000)
*
   HP_OPEN(IF(M_out=0,"SCREEN",IF(M_out=1,"PLOTTER","FILE")))
   PLOT()
   IF M_out=0
      INKEY(0)
   ENDIF
*
   TEXTMODE()
ENDDO

D:\FLIPPER>
D:\FLIPPER>
D:\FLIPPER>
```